

University of Groningen

## Active Control of Sound based on Diagonal Recurrent Neural Network

Jayawardhana, Bayu; Xie, Lihua; Yuan, Shuqing

*Published in:*  
Proceedings of the 41st SICE Annual Conference

**IMPORTANT NOTE:** You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

*Document Version*  
Publisher's PDF, also known as Version of record

*Publication date:*  
2002

[Link to publication in University of Groningen/UMCG research database](#)

*Citation for published version (APA):*

Jayawardhana, B., Xie, L., & Yuan, S. (2002). Active Control of Sound based on Diagonal Recurrent Neural Network. In *Proceedings of the 41st SICE Annual Conference* (Vol. 5, pp. 2666-2671). University of Groningen, Research Institute of Technology and Management.

**Copyright**

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

**Take-down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

*Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.*

# Active Control of Sound based on Diagonal Recurrent Neural Network

Bayu Jayawardhana<sup>1</sup>, Lihua Xie, Shuqing Yuan

School of EEE, Nanyang Technological University, Singapore 629798

<sup>1</sup> PK655853@ntu.edu.sg

**Abstract:** Recurrent neural network has been known for its dynamic mapping and better suited for nonlinear dynamical system. Nonlinear controller may be needed in cases where the actuators exhibit the nonlinear characteristics, or in cases when the structure to be controlled exhibits nonlinear behavior. The feedforward network with static characteristic usually uses a tapped delay input to control a nonlinear dynamic system. In the recurrent network, on the other hand, the dynamic behavior of the nonlinear system can be captured by the internal loop in its neurons and thus, a better system estimation and control can be expected using this control structure. In this paper, a multilayer perceptron diagonal recurrent neural network (DRNN) based control structure is employed to improve the performance of feedforward structure for Active Noise Control (ANC) systems where the nonlinearity occurs in the actuators. A comparison of DRNN with feedforward network is presented to highlight the improvement made by the recurrent structure.

**Keywords:** Active control of sound and vibration, recurrent neural network, nonlinear control

## I. Introduction

Active control of sound involves the introduction of a number of controlled "secondary" sources driven so that the field generated by these sources interferes destructively, 180° out of phase, with the noise field caused by the original "primary" source. Numerous works have been carried out for the ANC implementation under various environments: airplanes, auto vehicles, room acoustics, etc, see <sup>2,4,5</sup> and the references therein. Many adaptive linear filtering algorithms have been derived and proposed for the active sound control application in recent years <sup>2,4,5,13</sup>. One commonly used approach is the Normalized Filtered-x Least-Mean-Square (Normalized Fx-LMS) utilizing FIR filter. We will use the normalized Fx-LMS as a benchmark for the linear filter performance under non-linearity condition. Interest readers can refer to <sup>4,5</sup>.

In the cases where nonlinear characteristics are induced in the system, this linear controller may not perform well. The actuators, which consist of loudspeakers generated by amplifier, have a nominal value within which the system will remain in the linear region. It will develop into nonlinear characteristic when the input signal exceeds this nominal value, or when it operates below the minimum operating frequency of the actuator. Other possibility of nonlinear source is from the structure that inherits a nonlinear behavior. The performance of the linear filter as controller under these nonlinear circumstances may be degraded. Therefore, a nonlinear controller is preferred in this case. For the active control of sound and vibration, the use of neural networks as nonlinear control structures has been studied in <sup>1,9</sup>.

Bouchard *et. al.* <sup>1,9</sup> explored the multilayer perceptron neural networks using backpropagation scheme to control nonlinear plants, for the active control of sound. It uses the feedforward structure, combined with tapped delays,

and the backpropagation training algorithm <sup>7</sup> to solve the dynamical problems. However, the feedforward network can be considered as a static mapping network (if the tapped delay lines are omitted). On the other hand, recurrent neural networks have the ability to deal with time-varying systems through their own natural temporal operation <sup>3,6,7</sup>. Using the dynamic characteristic of recurrent neural network, the number of neurons required for controller or plant model can be reduced, and it is better suited for dynamical systems than the feedforward network <sup>6,7</sup>. This paper extends the DRNN algorithm for ANC and introduces new heuristic algorithms for DRNN, which have already been proposed for the feedforward network <sup>1,8,9</sup>, using Extended Kalman Filter (EKF) in improving the convergence rate.

With the same technique as the feedforward structure, we use two multilayer perceptrons for recurrent neural networks control-based structure, consisting of controller network and plant model network. Fig. 1 depicts the block diagram of recurrent neural network based control system.

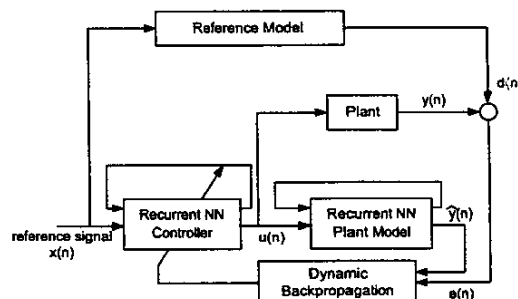


Fig. 1. Block diagram of recurrent neural network control based system

Bouchard and Snyder <sup>1,9</sup> gave a detailed description of feedforward neural network control structure. They use an

algorithm named Fx-BP that is a generalization from the filtered-x LMS algorithm, and is extended by Bouchard<sup>1)</sup> which introduced several improved training algorithms such as the so-called adjoint-BP, adjoint-EBP, adjoint-MEKA, adjoint-NEKA and adjoint-EKF (these methods employ the optimal filtering algorithms proposed by Shah<sup>8)</sup>). The adjoint approach can be formulated as follows:

*Adjoint approach:*

$$\text{instantaneous gradient} = \sum_{i=0}^L \frac{\partial e^2(n-L+i)}{\partial w_k(n-L)} \quad (1)$$

It uses the paradigm that the controller weights at a time instant affect the present and the next  $L$  samples of the cost function (quadratic error) or can be reformulated in causal way: that the weights of the control network at time instant  $n-L$  are determined by the cost function of the present and the last  $L$  samples, ( $e^2(n-L)$ ,  $e^2(n-L+1)$ , ...,  $e^2(n)$ ). However, the responsiveness of the adjoint algorithm has a tradeoff with the tapped delay length, since the present error gradient updates the weights in the past  $L$  weights. The longer the tapped delay  $L$ , the less responsive the adjoint algorithm becomes.

## II. Diagonal Recurrent Network Control Based

The diagonal configuration proposed by Ku et. al.<sup>3)</sup> for recurrent network provides a simple learning method, compared with the fully-connected recurrent structure, while preserving the dynamic capabilities of recurrent network. In diagonal recurrent network, each state output is fed back into its own state. The DRNN gradient-based learning method can be found in<sup>3)</sup>, and we will employ this configuration in the ANC application. Diagonal structure of recurrent network is illustrated in Fig. 2.

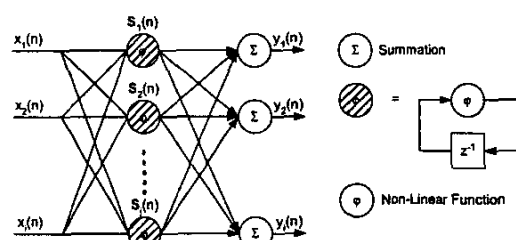


Fig. 2. Diagonal recurrent network layer

The DRNN control-based structure in<sup>3)</sup> does not employ tapped-delay structure in the input layer of plant model network, and it assumes that the system dynamic can be satisfied by the existence of the diagonal synapses. In our application, we still use the tapped delay in the controller and plant model structure for representing the unidentified zeros of the system by the diagonal synapses. Thus, the adjoint approach is utilized for control network learning

algorithm. For convenience and simplicity, we employ single channel ANC where the output layer of control network consists of one neuron, which corresponds to only one actuator. The extension to the multichannel algorithm is quite straightforward, and interested readers should refer to<sup>9)</sup>. The equations for the forward propagation in the multilayer perceptron DRNN are given below:

- For the input layer of the control network  
 $y_j^0(n) = x(n-j)$
- For the other layers of the control network  
 $s_j^m(n) = \left( \sum_i w_{i,j}^m(n) * y_i^{m-1}(n) \right) + w_{D,j}^m(n) * s_j^m(n-1)$   
 $y_j^m(n) = f(s_j^m(n))$
- For the input layer of the plant model network  
 $z_j^0(n) = y^M(n-j)$
- For the other layers of the plant model network  
 $t_j^m(n) = \left( \sum_i h_{i,j}^m * z_i^{m-1}(n) \right) + h_{D,j}^m * t_j^m(n-1)$   
 $z_j^m(n) = f(t_j^m(n))$  (2)
- For the output layer of the plant model  
 $e(n) = d(n) + z^M(n)$  (3)

where

- $x(n-j)$  signal from the reference sensor at time  $n-j$ ;
- $y_j^m(n)$  output of neuron  $j$  in layer  $m$  of the control network at time  $n$  ( $m > 0$ );
- $s_j^m(n)$  weighted sum of inputs for neuron  $j$  of layer  $m$  of the control network at time  $n$ ;
- $w_{i,j}^m(n)$  value at time  $n$  of the weight linking neuron  $i$  of layer  $m-1$  to neuron  $j$  of layer  $m$  of the control network;
- $w_{D,j}^m(n)$  value at time  $n$  of the self-loop weight (*diagonal weight*) in the neuron  $j$  of layer  $m$  of the control network;
- $z_j^m(n)$  output of neuron  $j$  in layer  $m$  of the plant model network at time  $n$ ;
- $t_j^m(n)$  weighted sum of inputs for neuron  $j$  of layer  $m$  of the plant model network at time  $n$ ;
- $h_{i,j}^m$  value of the weight linking neuron  $i$  of layer  $m-1$  to neuron  $j$  of layer  $m$  of the plant model network;
- $h_{D,j}^m$  value at time  $n$  of the self-loop weight (*diagonal weight*) in the neuron  $j$  of layer  $m$  of the control network;
- $e(n)$  error signal at time  $n$ ;
- $d(n)$  disturbance signal at time  $n$ ;

$M$  index of an output layer; number of layers in either control network or plant model network

From equation (2), there is additional term, compared with the feedforward networks<sup>1)</sup>, that contains feedback from the output neuron back into its own input junction, in the hidden layer. This is the term that offers dynamical modeling or better control of non-linear dynamic system than the feedforward networks. The activation function  $f(\cdot)$  is usually a nonlinear function for all neurons except those in the output layer. The activation functions are formulated as follows:

$f[x(n)] = x(n)$  for neurons in the output layer of control network and plant model network  
 $f[x(n)] = \tanh(a * x(n))$  for all other neurons (with  $a$  = arbitrary value)

The identification stage for the plant model network using diagonal recurrent structure can be referred to<sup>3)</sup>. We use the adjoint approach, as described in equation (1), in developing the learning algorithm of DRNN for ANC. The algorithm for the controller network using adjoint DRNN is summarized as follows:

- For the last layer of the plant model network  
 $\delta^M(n) = f'(t^M(n)) \cdot e(n) = e(n)$
- For the other layers of the plant model network (except the input layer)  
 $\delta_j^m(n) = f'(t_j^m(n)) \cdot \sum_k \delta_k^{m+1}(n) h_{j,k}^{m+1}$
- For the last layer of the control network  
 $\Delta^M(n-L) = f'(s^M(n-L)) \cdot \sum_k \sum_{l=0}^L \delta_k^1(n-L+l) h_{l,k}^1$   
 $= \sum_k \sum_{l=0}^L \delta_k^1(n-L+l) h_{l,k}^1$
- For the other layers of the control network (except the input layer)  
 $\Delta_j^m(n-L) = f'(s_j^m(n-L)) \cdot \sum_k \Delta_k^{m+1}(n-L) w_{j,k}^{m+1}(n-L)$   
 $\Psi_{i,j}^m(n-L) = R_{i,j}^m(n-L) \cdot \sum_k \Delta_k^{m+1}(n-L) w_{j,k}^{m+1}(n-L)$   
 $\Phi_{D,j}^m(n-L) = R_{D,j}^m(n-L) \cdot \sum_k \Delta_k^{m+1}(n-L) w_{j,k}^{m+1}(n-L)$

where

$$R_{i,j}^m(n-L) = \frac{\partial y_j^m(n-L)}{\partial w_{i,j}^m(n-L)}$$

$$= f'(s_j^m(n-L)) (y_i^{m-1}(n-L) + w_{D,j}^m(n-L) R_{i,j}^m(n-L-1))$$

and

$$R_{D,j}^m(n-L) = \frac{\partial y_j^m(n-L)}{\partial w_{D,j}^m(n-L)}$$

$$= f'(s_j^m(n-L)) (y_i^m(n-L-1) + w_{D,j}^m(n-L) R_{D,j}^m(n-L-1))$$

- For all the weights for the control network in hidden layer

$$w_{i,j}^m(n+1) = w_{i,j}^m(n) - \mu_i^m(n-L) \cdot \Psi_{i,j}^m(n-L)$$

$$\text{and } w_{D,j}^m(n+1) = w_{D,j}^m(n) - \mu_D^m(n-L) \cdot \Phi_{D,j}^m(n-L)$$

$$m = 1, 2, \dots, M-1 \quad (4)$$

For all the weights for the control network in output layer

$$w_{i,j}^M(n+1) = w_{i,j}^M(n) - \mu_j^M(n-L) \cdot y_i^{M-1}(n-L) \cdot \Delta^M(n-L)$$

where

$\delta_j^m(n)$  gradient of the instantaneous quadratic error relative to  $t_j^m(n)$ ;

$\Delta_j^m(n-L)$  sum of the gradients of the  $L+1$  past instantaneous quadratic errors relative to  $s_j^m(n-L)$ ;

$\Psi_{i,j}^m(n-L)$  the gradient of instantaneous quadratic error relative to  $w_{i,j}^m$

$\Phi_{D,j}^m(n-L)$  the gradient of instantaneous quadratic error relative to  $w_{D,j}^m$

$w_{i,j}^m(n)$  value at time  $n$  of the weight linking neuron  $i$  of layer  $m-1$  to neuron  $j$  of layer  $m$  of the control network;

$w_{D,j}^m(n)$  value at time  $n$  of the self-loop weight (diagonal weight) in the neuron  $j$  of layer  $m$  of the control network;

$\mu(n)$  adaptive learning rate;

$L$  number of delays in the tapped delay line between the control network and the plant model network

Different from equation of adjoint BP<sup>1)</sup>, in the adjoint DRNN, the effect from the past output information in each neurons has to be accounted via the diagonal recurrent weight  $w_{D,j}^m(n)$  (controller network) and  $h_{D,j}^m$  (plant model network). This increases the computational requirement of the adjoint DRNN about  $3(M+1)N$  multiplications from that of the adjoint BP, using the same neurons configuration, where  $M$  is the number of weights connected to the hidden neurons and  $N$  is the number of hidden neurons.

**Lemma 1.** For the DRNN learning algorithm, described in equation (4), the convergence is guaranteed if  $\mu(n)$  is chosen such that

$$0 < \mu(n) < \frac{2}{S_{\max}^2 g_{\text{norm}}^2(n)} \quad (5)$$

where  $g_{norm} = \|g(n)\|$ ;  $g(n) = \frac{\partial y^M(n)}{\partial w(n)}$ ;

$S_{max} = \max_n \|S(n)\|$ ;  $\| \cdot \|$  is the euclidian norm;

*Proof:* See <sup>3)</sup>

**Lemma 2.** For the multilayer perceptron DRNN structure, used in equation (2), the sensitivity of the plant model  $S(n)$ , defined as:

$$S(n) = \frac{\partial z^M(n)}{\partial y^M(n-L)} = \sum_i^L \sum_j^{N_i^1} h_{i,j}^1 \delta_j^1(n+i-L); \quad (6)$$

with  $\delta_j^m(n)$  is the instantaneous gradient in the layer  $m$  of plant model network and  $N_i^m$  is the number of neurons in layer  $m$  of plant model network, can be upper-bounded by:

$$\|S(n)\| \leq \frac{L}{a} \prod_{n=1}^M N_i^n \cdot a \cdot h_{i,j}^{n,max} \equiv S_{max} \quad (7)$$

where  $h_{i,j}^{n,max} = \max_{i,j} |h_{i,j}^n|$ ;  $a$  is the scaling factor in the nonlinear function  $f(\cdot)$  and  $L$  is the tapped-delay length of the input layer in plant model.

*Proof:* It is straightforward by backpropagating the sensitivity from the outer layer of plant model network to the input layer.

**Remark 1.** Let  $\mu_I^M(n)$ ,  $\mu_I^m(n)$ ,  $\mu_D^m(n)$  be the adaptive learning rate for the controller weights  $w_{i,j}^M(n)$ ,  $w_{i,j}^m(n)$  and  $w_{D,j}^m(n)$ , respectively. Then the Adjoint DRNN learning algorithm convergence of (4) is guaranteed if the adaptive learning rates are bounded by the following criteria. It is based on the Lemma 1, where the sensitivity  $S_{max}$  is obtained from Lemma 2.

For the controller weights connected to the output layer:

$$g_{norm}^M(n) = \left\| \frac{\partial y^M(n)}{\partial w^M(n)} \right\| = \|y^{M-1}(n)\|$$

where  $\| \cdot \|$  is the euclidian norm and  $y^{M-1}(n)$  as described in (4). The adaptive learning rate is bounded by

$$0 < \mu_I^M(n) < \frac{2}{S_{max}^2 (g_{norm}^M(n))^2} \quad (8)$$

For the controller weights connected to the hidden layer:

$$g_{norm}^m(n) = \left\| \frac{\partial y^M(n)}{\partial w^m(n)} \right\|; \text{ where } \| \cdot \| \text{ is the euclidian norm}$$

$$\frac{\partial y^M(n)}{\partial w^m(n)} = [g_1^m(n) \quad g_j^m(n) \quad \cdots \quad g_J^m(n)]^T$$

$$g_j^m = [g_{i,j}^m(n) \quad g_{i,j}^m(n) \quad \cdots \quad g_{i,j}^m(n) \quad g_{D,j}^m(n)]^T$$

$$\text{and } g_{i,j}^m(n) = \frac{\partial y^M(n)}{\partial w_{i,j}^m(n)}; \quad g_{D,j}^m(n) = \frac{\partial y^M(n)}{\partial w_{D,j}^m(n)};$$

Then, the adaptive learning rate is bounded by

$$0 < \mu_I^m(n) < \frac{2}{S_{max}^2 (g_{norm}^m(n))^2} \quad (9)$$

$$\text{and } 0 < \mu_D^m(n) < \frac{2}{S_{max}^2 (g_{norm}^m(n))^2} \quad (10)$$

where  $g_{i,j}^m(n)$  and  $g_{D,j}^m(n)$  are the sensitivity of control network, from the output layer of control network to the weight in the hidden layer of control network, and are calculated in a similar fashion of (4) using backpropagation technique.

**Remark 3.** The optimal convergence rate, as the controller case described in <sup>3)</sup>, for the learning algorithm above is:

$$\mu_I^M(n) = \frac{1}{S_{max}^2 (g_{norm}^M(n))^2} \quad (11)$$

$$\mu_I^m(n) = \frac{1}{S_{max}^2 (g_{norm}^m(n))^2} \quad (12)$$

$$\text{and } \mu_D^m(n) = \frac{1}{S_{max}^2 (g_{norm}^m(n))^2} \quad (13)$$

which is the half of the upper limit in (8), (9) and (10).

The gradient descent method above offers a simple training method and low storage requirement, but inherits slow convergence rate and is a non-optimal solution. The introduction of non-linear recursive-least-square for neural network <sup>8)</sup>, fosters the development of an improved algorithm, with better quadratic error performance and faster convergence rate, but in the cost of higher computational complexity. We extend the adjoint algorithm above using EKF for improving the convergence rate, and also applying the NEKA (Neuron-level Extended Kalman Algorithm) and MEKA (Multiple Extended Kalman Algorithm) of <sup>8)</sup> into DRNN in reducing the high computational requirement of EKF. In the simulation result, we also present the result of the DRNN based on this nonlinear recursive-least-square algorithm, though the complete algorithm is not described in this paper.

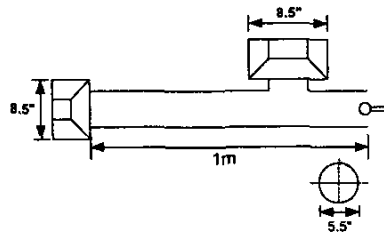
Among the learning algorithms that have been discussed in this paper, the adjoint EKF algorithm for DRNN is the most demanding algorithm in computation as can be observed from Table 1 below. However, since EKF accounted the global model of the network, the resulting EKF DRNN is expected to achieve the best performance from the other two algorithms.

**Table 1. Computational Complexity of Adjoint-DRNN algorithms, with  $M$  is the number of weights per hidden neuron and  $N$  is the number of hidden neurons**

Algorithms	Order of multiplication
Adjoint DRNN	$O(MN)$
Adjoint EKF DRNN	$O((MN)^2)$
Adjoint NEKA DRNN	$O(M^2N)$
Adjoint MEKA DRNN	$O(M^2N)$

### III. Simulation and Experiment Result

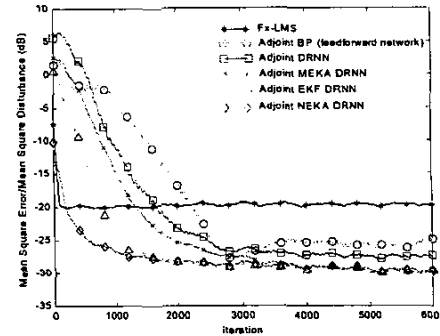
In examining the performance of nonlinear controller for active noise cancellation, a simulation based on the experimental narrow duct was carried out, where a hard excitation signal of 86.4Hz was introduced, then an experimental result was conducted using the parameter obtained from the simulation. The experimental duct configuration is illustrated in Fig. 3. The primary disturbance is generated from the speaker placed in the end of the duct, and the canceling loudspeaker is positioned at the side of the duct near the outlet. The sampling frequency used is 2kHz and it is ensured that no aliasing occurred in the system caused by the digital equipment.



**Fig. 3. Experimental narrow duct geometry**

The first five calculated duct modes, using the wave equation for the corresponding duct, are 86.4, 259.13, 431.87, 604.6 and 777.4 Hz. If we consider non-linear saturation behavior exists in the actuator, then theoretically, if we excite the first mode, 86.4Hz, the non-linear behavior of the actuator saturation will excite the harmonic series, including 259.12, 431.87, 604.6 and 777.4 Hz modes. Therefore, using the primary disturbance of 86.4Hz, the non-linear ANC aims to attenuate this 1<sup>st</sup> mode frequency while minimizing the effect of the non-linear behavior of actuator, especially those related to the duct modes.

The DRNN neuron configuration for the plant model network uses 40-40-1, corresponding to 40 tapped-delay neurons in the input layer, 40 diagonal neurons in the hidden layer and 1 neuron in the output layer. And for later uses in the comparison, the feedforward network plant model also uses 40-40-1 neuron configuration. The identification gives a good result, especially those pertaining to the frequency of interests. The control network neuron configuration uses multilayer perceptrons of 20-20-5-1, for both DRNN or the feedforward network. The simulation result is presented in Fig. 4.



**Fig. 4. Simulation result of non-linear ANC using DRNN, Feedforward Network and Linear Filter**

From several simulations and different weight initialization, the DRNN algorithm performed better than the feedforward network, though we do not compare with the nonlinear recursive-least-square version of the feedforward network. And as expected, the non-linear neural network was superior to the linear filtering, where the linear filtering only reduced the primary frequency without concerning with the harmonics series, 19dB was achieved by the linear filter and more than 26dB was obtained by the neural network (Table 2). In overall, the DRNN structure yields better performance than the feedforward network for ANC problem for the simulation result.

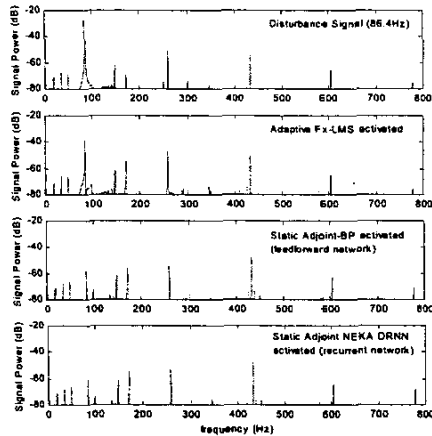
**Table 2. Error performance and computational requirement (based on Matlab® flops computation) of non-linear ANC using DRNN**

Algorithm	Total Noise Reduction (dB)	Flops Requirement
Normalized Fx-LMS	19.7346	1,927
Adjoint-BP	26.2102	10,794
Adjoint-DRNN	27.7685	14,050
Adjoint-MEKA DRNN	29.7746	70,152
Adjoint-NEKA DRNN	30.0938	70,676
Adjoint-EKF DRNN	29.8077	1,982,182

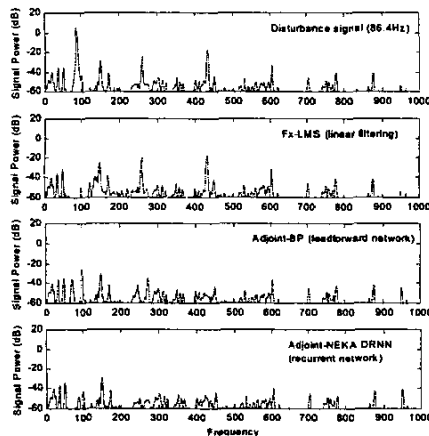
Fig. 5 below shows the experimentation result, where we can see that the nonlinear controller based on neural networks worked better than the linear adaptive filtering, in dealing with non-linearity. About 7.5dB improvement can be achieved using the neural network from the adaptive linear filter. As a comparison, Fig. 5(b) shows the simulation result from which a close observation indicates that NEKA DRNN attenuates the primary disturbance and the harmonics better than the other algorithm (Table 2). During the experiments, this superiority of NEKA with the feedforward network is not so obvious as the one shown from simulation result.

**Table 3. ANC experimental result using static controller**

Algorithm	Total Energy Reduction (dB)
Normalized Fx-LMS	9.9755
Adjoint BP	17.4335
Adjoint NEKA DRNN	17.6610



(a)



(b)

Fig. 5. (a) non-linear ANC experimental result using static neural network controller, and adaptive linear controller; (b) non-linear ANC simulation result using neural network and linear filtering

## Conclusions

New improved heuristic adaptive training algorithms for DRNN control-based structure are introduced with its application to the ANC problem. The recurrent structure of DRNN has better ability in dealing with non-linear dynamic system than the feedforward structure. Simulation and

experimental results verified that the overall performance of DRNN is better than that of the feedforward network.

## References

- [1] Bouchard, M., Paillard, B., Dinh, C.T.L., "Improved Training of Neural Networks for the Nonlinear Active Control of Sound and Vibration", IEEE Trans. on Neural Networks, Vol. 10, pp. 391-401, March 1999
- [2] Gelin, L.J., "Active Noise Control: A Tutorial for HVAC Designer", ASHRAE Journal, pp. 43-49, August 1997.
- [3] Ku, C.C., Lee, K.Y., "Diagonal Recurrent Neural Networks for Dynamic Systems Control", IEEE Trans. on Neural Networks, vol. 6, pp. 144-156, January 1995.
- [4] Kuo, S.M., Morgan, D.R., "Active Noise Control Systems-Algorithms and DSP Implementations", New York: Wiley, 1996.
- [5] Nelson, P.A., Elliot, S.J., "Active Control of Sound", London, U.K.: Academic, 1992.
- [6] Nerrand, O., Roussel-Ragot, P., Urbani, D., Personnaz, L., Dreyfus, G., "Training Recurrent Neural Networks: Why and How? An Illustration in Dynamical Process Modeling", IEEE Trans. on Neural Networks, Vol. 5, pp. 178-184, March 1994.
- [7] S. Haykin, "Neural Networks: A Comprehensive Foundation", New Jersey: Macmillan, 1994.
- [8] Shah, S., Palmieri, F., Datum, M., "Optimal Filtering Algorithms for Fast Learning in Feedforward Neural Networks", Neural Networks, Vol. 5, pp. 779-787, 1992.
- [9] Snyder, S.D., Tanaka, N., "Active Control of Vibration Using a Neural Network", IEE Trans. on Neural Networks, vol. 6, pp. 819-828, July 1995.
- [10] Hassibi, B., Sayed, A.H., Kailath, T., "H<sup>∞</sup> Optimality of the LMS Algorithm", IEEE Trans. on Signal Processing, Vol. 44, pp. 267-280, February 1996.
- [11] Bouchard, M., Quednau, S., "Multichannel Recursive-Least-Squares Algorithms and Fast-Transversal-Filter Algorithms for Active Noise Control and Sound Reproduction Systems", IEEE Trans. on Speech and Audio Processing, Vol. 8, pp. 606-618, September 2000.
- [12] Haykin, S., "Adaptive Filter Theory", 3<sup>rd</sup> ed., New Jersey: Prentice-Hall, 1996.
- [13] Douglas, S., "Fast, Exact Filtered-x LMS and LMS Algorithms for Multichannel Active Noise Control", Proc. ICASSP-97, Munich, Germany 1997.